

Rancang Bangun Aplikasi Simulasi Jaringan untuk Praktikum Jaringan Komputer Memanfaatkan Scapy dalam Lingkungan LXC

Dhimas Bagus Pramudya, Royyana Muslim Ijtihadie, dan Baskoro Adi Pratomo
Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: roy@its-sby.edu

Abstrak—Praktikum Jaringan Komputer yang sudah berlangsung di Jurusan Teknik Informatika dalam menyimulasikan topologi jaringan menggunakan User Mode Linux (UML). Namun server mengalami penurunan kinerja, ketika menyimulasikan topologi dalam jumlah banyak yang menggunakan host dan *router*. Hal tersebut mengakibatkan waktu startup yang lama, latency yang tinggi, dan kecenderungan error sering dialami server.

Linux Container merupakan lingkungan virtual berbasis container yang menyediakan ukuran *image* yang lebih kecil dari User Mode Linux. Proses startup pada Linux Container lebih cepat dibandingkan User Mode Linux meskipun banyak aplikasi server yang sudah dipasang. Dengan kelebihan Linux Container diatas, kecenderungan error yang diakibatkan oleh User Mode Linux dapat dikurangi. Selain itu dalam praktikum jaringan komputer manipulasi paket dengan *scapy* dapat dimanfaatkan untuk memperdalam praktikan dalam menganalisa paket secara detail apa saja yang dikirim dan respon yang diberikan oleh paket tersebut. Oleh karena itu dalam Tugas Akhir ini diusulkan lingkungan virtualisasi yaitu Linux Container (LXC) dan simulasi pengiriman paket menggunakan Scapy.

Dari hasil uji coba dengan sistem simulasi jaringan ini, didapatkan waktu respon untuk menangani setiap rute sekitar kurang dari satu detik sampai tujuh detik. Komputer *web server* yang digunakan mampu membuat virtualisasi sebanyak 400 dengan *web server* yang sedang diakses oleh 20 pengguna.kata.

Kata Kunci— Praktikum, Simulasi Jaringan, Linux Container, Scapy

menggunakan host dan *router*, server mengalami penurunan kinerja. Selain itu sumber daya penyimpanan di server juga berkurang karena besarnya ukuran *image* yang dibuat oleh UML untuk masing- masing host dan *router*. Hal ini dapat menyebabkan beberapa masalah seperti waktu startup yang lama, latency yang tinggi, dan kecenderungan error.

Makalah ini membahas tentang rancang bangun aplikasi simulasi jaringan untuk praktikum jaringan komputer di Teknik Informatika ITS memanfaatkan Scapy dalam lingkungan LXC. Linux Container merupakan lingkungan virtual berbasis container yang menyediakan ukuran *image* yang lebih kecil dari User Mode Linux [1]. Proses startup pada Linux Container lebih cepat dibandingkan User Mode Linux meskipun banyak aplikasi server yang sudah dipasang. Dengan kelebihan Linux Container diatas, kecenderungan error yang diakibatkan oleh User Mode Linux dapat dikurangi.

Scapy [2] merupakan program untuk memanipulasi paket. Scapy dapat menata ulang sebuah paket dari beberapa protokol, mengirimkannya melalui jaringan, menganalisa paket tersebut, mencocokkan antara request dan reply . Scapy juga dapat digunakan untuk scanning, tracerouting, probing, unit tests, penyerangan atau network discovery. Sehingga dalam praktikum jaringan komputer, Scapy dapat dimanfaatkan untuk memperdalam praktikan dalam menganalisa paket secara detail apa saja yang dikirim dan respon yang diberikan oleh paket tersebut.

I. PENDAHULUAN

PRAKTIKUM jaringan komputer merupakan kegiatan untuk menunjang mata kuliah jaringan komputer. Dalam praktikum jaringan komputer, praktikan diberi panduan untuk membuat simulasi dari jaringan komputer yang sebenarnya. Simulasi jaringan dalam praktikum jaringan komputer merupakan metode bagi praktikan untuk membuat topologi jaringan komputer tanpa harus memakai sumber daya yang sesungguhnya. Praktikum Jaringan Komputer yang sudah berlangsung di Jurusan Teknik Informatika dalam menyimulasikan topologi jaringan menggunakan User Mode Linux (UML). User Mode Linux merupakan aplikasi yang digunakan untuk menyimulasikan *router*, host dan *switch* dalam lingkungan virtual.

User Mode Linux mempunyai beberapa kekurangan, yaitu ketika menyimulasikan topologi dalam jumlah banyak yang

II. TINJAUAN PUSTAKA

A. Simulasi Jaringan

Simulasi jaringan [3] merupakan virtual reality simulation yang digunakan untuk menirukan tabiat dari proses dan sistem jaringan sesuai dengan dunia nyata. Berbagai percobaan dapat dilakukan dengan mengubah model pada simulasi. Penggunaan simulasi dapat membantu untuk menguji hal yang terlalu beresiko jika dilakukan secara nyata.

B. Linux Container(LXC)

Linux Container (LXC) [4] adalah sebuah lingkungan virtualisasi dilevel sistem operasi yang dapat menjalankan sistem Linux standar yang terisolasi dalam sebuah host tanpa terpisah oleh kernel host tersebut. Tabel 1 merupakan daftar

fitur yang diakomodasi oleh Linux Container.

Tabel 1. Daftar Fitur Linux Container

No	Nama Fitur	Keterangan
1	<i>Kernel namespaces(inter process communication, mount, pid, network, dan user)</i>	Virtualisasi yang dibuat oleh Linux Container memiliki lingkungan isolasi sendiri berbeda dengan komputer <i>host</i> , sehingga memiliki proses, jaringan, <i>user</i> , dan <i>inter process communication</i> yang berbeda dengan komputer <i>host</i> .
2	Apparmor dan SELinux <i>profiles</i>	Fitur keamanan yang disediakan Linux Container.
3	Seccomp <i>policies</i>	Merupakan fitur dari <i>kernel</i> yang dapat menyaring <i>system call</i> yang biasa digunakan oleh proses dalam sistem operasi Linux beserta anak dari proses tersebut. Seccomp <i>policies</i> dapat ditambahkan pada virtualisasi oleh Linux Container.
4	CGroups	Pengguna dapat membatasi dan mengatur prioritas penggunaan sumber daya seperti <i>cpu</i> , <i>memory</i> , <i>block I/O</i> , dan <i>network</i> .

Linux Container terdiri dari beberapa komponen pengembangannya, antara lain liblxc sebagai pustaka khusus untuk Linux Container dan bahasa pemrograman yang digunakan untuk API antara lain python3, lua, go, dan ruby. Selain itu Linux Container juga menyediakan standar dalam mengontrol container-nya dan template image untuk container.

Tabel 2 merupakan daftar perintah dari Linux Container [5] yang digunakan dalam tugas akhir ini.

C. Scapy

Scapy [2] adalah program untuk memanipulasi sebuah paket. Scapy dapat menata ulang sebuah paket dari beberapa protokol, mengirimkannya melalui jaringan, menganalisa paket tersebut, dan mencocokkan antara *request* dan *reply*.

Scapy dapat digunakan untuk *scanning*, *tracerouting*, *probing*, *unit tests*, penyerangan atau *network discovery*. Scapy dibangun dengan bahasa pemrograman python.

D. Flask

Flask [6] merupakan kerangka kerja mikro dari bahasa pemrograman python yang didukung oleh *werkzeug* dan *jinja2*. Aplikasi web yang dibuat dengan Flask disimpan dalam satu berkas *.py*.

Flask ingin menjadi web framework yang sederhana namun dapat diperluas dengan beragam pustaka tambahan yang sesuai dengan kebutuhan penggunaanya.

E. GoJS

GoJS [7] merupakan pustaka dari Javascript untuk mengimplementasikan diagram interaktif. Diagram yang mempunyai konstruksi *node* dan penghubung antar *node* yang sulit sekalipun dapat dibuat menggunakan GoJS dengan mudah karena mempunyai *template* dan *layout* yang beragam. Selain itu, fitur yang didukung oleh GoJS antara lain *drag and drop*, *copy and paste*, *data binding and models*, *transactional state* and *undo management*, *palettes*, *overviews*, dan *event handlers*.

GoJS digunakan sebagai tampilan untuk simulasi dalam tugas akhir ini. Fitur GoJS yang dimanfaatkan dalam tugas akhir ini antara lain *template*, *data binding and models*, *drag and drop*, *palettes* dan *event handlers*.

Tabel 2. Daftar Perintah LXC

No	Perintah	Contoh penggunaan	Keterangan
1	<i>lxc-create</i>	<i>lxc-create -n [nama kontainer] -t [nama template sistem operasi]</i>	Membuat kontainer baru.
2	<i>lxc-clone</i>	<i>lxc-clone -n [nama kontainer baru] -o [nama kontainer yang ingin disalin]</i>	Menyalin kontainer yang sudah tersedia.
3	<i>lxc-start</i>	<i>lxc-start -n [nama kontainer]</i>	Menghidupkan kontainer.
4	<i>lxc-stop</i>	<i>lxc-stop -n [nama kontainer]</i>	Mematikan kontainer.
5	<i>lxc-destroy</i>	<i>lxc-destroy -n [nama kontainer]</i>	Menghapus kontainer.
6	<i>lxc-ls</i>	<i>lxc-ls --fancy</i>	Menampilkan semua daftar kontainer yang sudah dibuat dengan statusnya.
7	<i>lxc-info</i>	<i>lxc-info -n [nama kontainer]</i>	Menampilkan informasi dari satu kontainer.

Tabel 3. Daftar Perintah Bridge-utils

No	Perintah	Keterangan
1	<i>brctl addbr [nama bridge]</i>	Menambahkan <i>bridge</i> baru.
2	<i>brctl addif [nama bridge] [interface]</i>	Menambahkan <i>interface ethernet</i> ke dalam <i>bridge</i> yang sudah dibuat sebelumnya.
3	<i>brctl show</i>	Menampilkan daftar semua <i>bridge</i> yang sudah dibuat beserta <i>interface</i> yang terdapat dalam <i>bridge</i> tersebut.
4	<i>ifconfig [nama bridge] up</i>	Menghidupkan <i>bridge</i> yang sudah dibuat
5	<i>ifconfig [nama bridge] down</i>	Mematikan <i>bridge</i> yang sudah dibuat
6	<i>ifconfig [nama bridge] [alamat ip] netmask [netmask] up</i>	Menghidupkan <i>bridge</i> dengan mengatur alamat <i>ip</i> dan <i>netmask</i> .

F. Bridge-utils

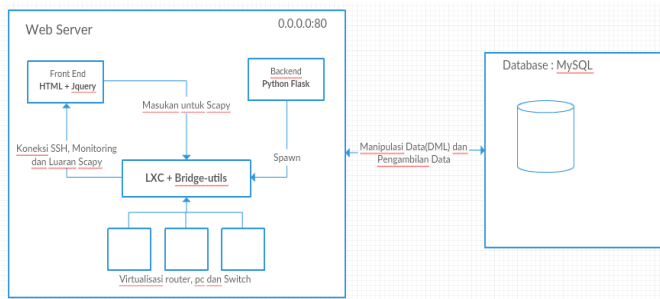
Bridge-utils [8] merupakan paket aplikasi yang digunakan untuk mengkonfigurasi *bridge* untuk *ethernet* di sistem operasi Linux. Bridge-utils dapat digunakan untuk menghubungkan perangkat *ethernet* sekaligus secara bersama-sama baik dari *host* yang mempunyai fisik nyata maupun berupa virtualisasi.

Dalam tugas akhir ini, Bridge-utils digunakan untuk menyimulasikan modul *switch* sesuai dengan keadaan di dunia nyata. Tabel 3 merupakan daftar perintah Bridge-utils.

III. ANALISA DAN PERANCANGAN

A. Deskripsi Fitur

Simulasi jaringan pada umumnya merupakan virtual reality simulation yang digunakan untuk menirukan tabiat dari proses dan sistem jaringan sesuai dengan dunia nyata. Secara umum,



Gambar 2. Desain Arsitektur Sistem

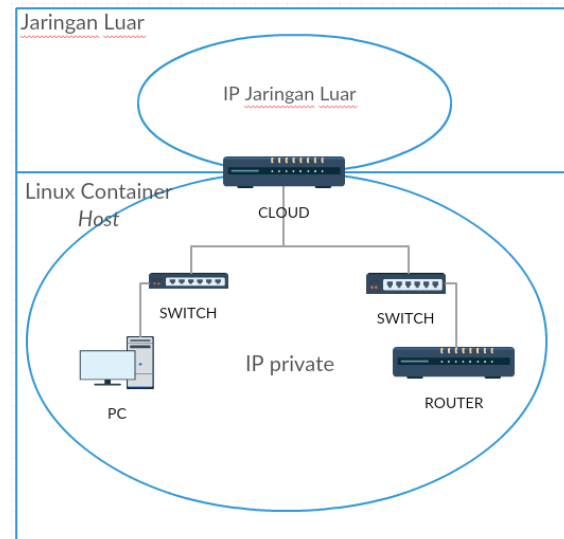
fitur sistem dibagi menjadi dua yaitu fitur untuk praktikan dan fitur untuk admin. Fitur yang tersedia untuk praktikan :

- Praktikan dapat menyalin template virtualisasi yang dibuat oleh admin.
- Praktikan dapat menghidupkan virtualisasi yang telah dibuat.
- Praktikan dapat mematikan virtualisasi yang telah dibuat.
- Praktikan dapat menghapus virtualisasi yang telah dibuat.
- Praktikan dapat mengakses virtualisasi yang telah dibuat menggunakan SSH.
- Praktikan dapat mengatur jaringan virtualisasi.
- Praktikan dapat mengubah konfigurasi jaringan virtualisasi yang sudah diatur sebelumnya.
- Praktikan dapat membuat simulasi paket dengan Scapy. Sedangkan fitur yang tersedia untuk admin:
- Admin dapat membuat template virtualisasi yang nantinya akan digunakan praktikan.
- Admin dapat menambahkan praktikan baru.
- Admin dapat menghapus praktikan.
- Admin dapat menginstal aplikasi pada template sesuai komponen.
- Admin dapat menghapus template virtualisasi yang telah dibuat.
- Admin dapat merubah pengaturan jaringan pada Linux Container.
- Admin dapat memilih template yang digunakan dalam simulasi.

B. Arsitektur Sistem

Gambar 2 merupakan desain arsitektur dari sistem yang akan dibuat. Sistem dibangun dalam beberapa komponen umum yang terkait satu sama lain, yaitu:

- *Backend* sebagai pengendali yang menghubungkan basis data dengan *frontend*.
- *Frontend* adalah antarmuka dari sistem yang berbasis web.
- Linux Container dan Bridge-utils sebagai lingkungan virtual yang dipanggil dengan bantuan *Backend*. Digunakan sebagai lingkungan untuk menyimulasikan perangkat jaringan seperti *router*, *switch*, komputer dan *cloud*.
- Basis data sebagai tempat menyimpan segala macam data transaksi maupun data utama sistem.
- Manipulasi paket dengan Scapy sebagai aplikasi yang dapat memodifikasi paket untuk simulasi pengiriman paket oleh praktikan.



Gambar 1. Desain Komponen Virtualisasi

C. Desain Backend dan Frontend

Backend dibuat dengan kerangka kerja dari bahasa python yaitu Flask. *Backend* akan menangani segala bentuk operasi basis data yang dikirimkan dari *frontend*, memberikan data yang dibutuhkan *frontend*, dan mengirim perintah ke Linux Container sesuai dengan data dari *frontend*. Terdapat tiga jenis rute yang ditangani oleh *Backend*, yaitu rute yang diakses tanpa autentikasi, rute yang hanya bisa diakses admin, dan rute yang hanya bisa diakses praktikan. Sedangkan bagian *frontend* menggunakan HTML dan JQuery sedang untuk tampilan simulasi menggunakan pustaka javascript GoJS.

D. Desain Linux Container dan Bridge-utils

Linux Container digunakan sebagai lingkungan virtualisasi. Komponen yang divirtualisasikan adalah *router*, komputer(PC), dan *cloud*. Admin akan membuat template untuk masing-masing komponen. Sedangkan ketika melakukan simulasi, praktikan akan melakukan penyalinan dari template komponen yang sudah disediakan. Sistem operasi untuk komponen – komponen tersebut adalah Ubuntu.

Komponen *cloud* digunakan untuk menghubungkan jaringan dalam lingkungan Linux Container dengan jaringan luar menggunakan virtual *switch*. Bridge-utils digunakan untuk virtual *switch* tersebut. Selain itu, Bridge-utils juga digunakan untuk menghubungkan komponen-komponen lain yaitu komputer dan *router*. Dalam komponen *cloud* diinstal aplikasi shellinabox, aplikasi yang dapat menampilkan terminal melalui peramban, untuk mengakses virtualisasi lain menggunakan koneksi SSH. Untuk melakukan simulasi Scapy, diinstal aplikasi Scapy di dalam komponen komputer.

Komponen komputer dan *router* dalam lingkungan Linux Container menggunakan alamat IP *private* sedangkan komponen *cloud* menggunakan IP yang terhubung dari jaringan luar dan IP *private* yang terhubung dengan jaringan di lingkungan Linux Container. Gambar 1 merupakan desain dari komponen virtualisasi yang akan diimplementasikan pada sistem.

E. Desain Manipulasi Paket dengan Scapy

Scapy adalah program untuk memanipulasi sebuah paket. Praktikan akan mendefinisikan paket terlebih dahulu melalui antarmuka web, kemudian sistem akan membuat sebuah file berekstensi .py yang berisi paket tersebut di dalam komponen virtual komputer. Praktikan dapat mensimulasikan pengiriman paket dengan Scapy melalui komponen virtual computer.

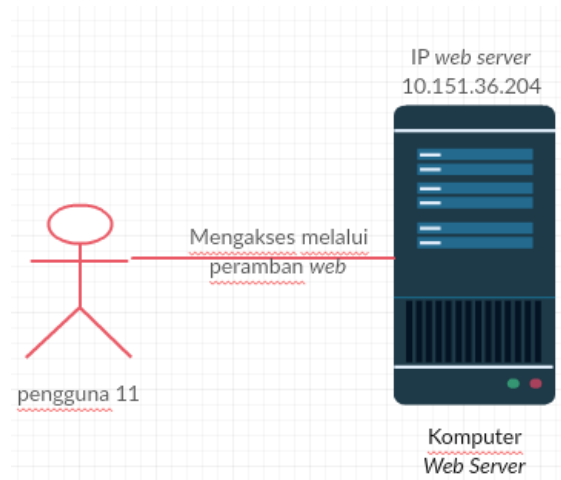
IV. PENGUJIAN DAN EVALUASI

A. Uji Fungsionalitas

Uji dilakukan dengan melakukan uji coba membuka aplikasi melalui peramban web. Gambar 3 merupakan desain arsitektur uji fungsionalitas. Uji coba ini berguna untuk menguji apakah operasi-operasi dasar yang dilakukan dapat memberikan hasil yang diharapkan. Peramban web yang digunakan adalah Mozilla Firefox dan fitur add-ons Firebug untuk melihat waktu respon yang diberikan. Tabel 4 merupakan waktu respon rata-rata uji fungsionalitas dalam lima kali percobaan.

Tabel 4. Waktu Respon Uji Fungsionalitas

No	Nama Kegiatan	Waktu Respon(ms)					Rata-rata
		1	2	3	4	5	
1	Login sebagai admin	765	881	788	120	840	894.8
2	Login sebagai praktikan	850	774	853	807	865	829.8
3	Menyalin template virtualisasi	993	217	193	337	725	4930
4	Menghidupkan virtualisasi	129	370	416	440	526	608.4
5	Mematikan virtualisasi	952	857	801	782	714	821.2
6	Menghapus virtualisasi	554	433	416	402	378	436.6
7	Mengakses virtualisasi dengan SSH	80	83	86	76	74	79.8
8	Mengatur jaringan virtualisasi	49	56	36	53	48	48.4
9	Mengubah konfigurasi jaringan virtualisasi	49	8	9	13	56	27
10	Membuat simulasi paket ICMP dengan Scapy	45	43	44	50	53	47
11	Membuat simulasi paket TCP dengan Scapy	50	47	45	42	49	46.6
12	Membuat template virtualisasi	383	406	738	772	435	5468
13	Menambahkan praktikan	69	67	61	74	79	70
14	Menghapus praktikan	51	55	52	52	54	52.8
15	Menginstal	598	646	519	394	386	5086



Gambar 3. Desain Arsitektur Uji Fungsionalitas

No	Nama Kegiatan	Waktu Respon(ms)					Rata-rata
		1	2	3	4	5	
	aplikasi pada template virtualisasi	0	0	0	0	0	
16	Menghapus template virtualisasi	458	414	392	375	359	399.6
17	Mengubah pengaturan jaringan Linux Container	118	99	111	116	113	111.4
18	Memilih template untuk simulasi	123	99	112	132	112	115.6

B. Uji Fungsionalitas dengan Skenario Modul Praktikum

Uji dilakukan oleh pengguna dari mahasiswa Teknik Informatika ITS dengan melakukan uji coba sesuai dengan modul praktikum jaringan komputer. Modul yang akan digunakan sebagai uji coba adalah modul *web server*. Modul tersebut membutuhkan sebuah komponen *cloud*, sebuah komponen komputer dan sebuah komponen *switch*.

Web server akan diinstal pada komponen *cloud* dan aplikasi peramban pada terminal yaitu lynx sudah terinstal pada komputer. *Web server* akan diakses melalui jaringan ajk dan jaringan dalam Linux Container. Ketika berhasil diakses maka modul *web server* berhasil dilaksanakan.

Pengujian menggunakan skenario modul praktikum jaringan komputer dilakukan oleh pengguna bernama Daniel Fablius dan Kharisma Nur Annisa. Tabel 5 merupakan hasil pengujian dengan skenario modul praktikum *web server*.

C. Uji Kapasitas dan Performa

Uji kapasitas dan performa terdiri dari dua pengujian, yaitu uji kapasitas dan performa pada komputer *web server* dan uji

Tabel 5. Hasil Uji Fungsionalitas dengan Skenario Modul Praktikum

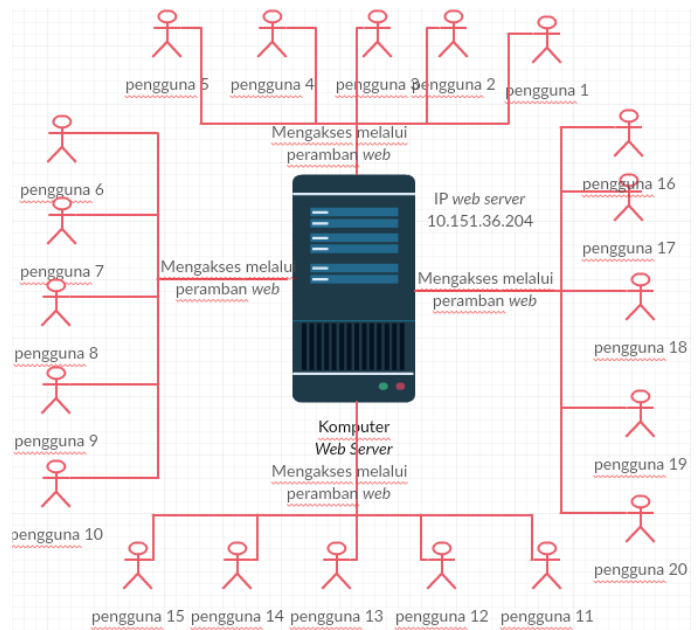
No	Nama Pengguna	NRP	Kegiatan	Keterangan dan Catatan
1	Daniel Fablius	5113100109	Menyalin komponen <i>cloud</i> ,	Berhasil

No	Nama Pengguna	NRP	Kegiatan	Keterangan dan Catatan
			switch, dan komputer.	
			Mengatur jaringan sesuai dengan modul.	Berhasil
			Melakukan ping dari cloud ke komputer.	Berhasil
			Menginstal aplikasi apache2 dan php5 pada komponen cloud.	Berhasil
			Mengakses cloud melalui komputer Lab AJK dengan peramban web.	Berhasil
			Mengakses cloud melalui komponen komputer menggunakan peramban web lynx	Berhasil
Catatan dari pengguna : Perlu ditambahkan cara penggunaan pada halaman simulasi				
2	Kharisma Nur Annisa	5113100026	Menyalin komponen cloud, switch, dan komputer.	Berhasil
			Mengatur jaringan sesuai dengan modul.	Berhasil
			Melakukan ping dari cloud ke komputer.	Berhasil
			Menginstal aplikasi apache2 dan php5 pada komponen cloud.	Berhasil
			Mengakses cloud melalui komputer Lab AJK dengan peramban web.	Berhasil
			Mengakses cloud melalui komponen komputer menggunakan peramban web lynx.	Berhasil
Catatan dari pengguna : tampilan koneksi dikelompokkan sesuai dengan komponen masing-masing.				

kapasitas pada komputer pembanding. Uji coba pada komputer web server dilakukan untuk menguji kemampuan web server dalam menangani simulasi ketika banyak virtualisasi berjalan pada komputer web server.

Uji coba pada komputer pembanding dilakukan untuk mengetahui spesifikasi minimum komputer untuk melakukan virtualisasi menggunakan Linux Container dan seberapa banyak virtualisasi yang dapat dibuat.

Uji kapasitas dan performa dilakukan dengan mengakses web server melalui peramban web pada komputer klien sebanyak sepuluh. Peramban web yang digunakan dimasing-masing komputer adalah Mozilla Firefox dan Chrome. Gambar 4 merupakan desain arsitektur uji coba kapasitas dan performa untuk komputer web server. Berikut langkah – langkahnya dalam proses pengujian dengan komputer web server :



Gambar 4. Desain Arsitektur Uji Kapasitas dan Performa Komputer Web Server

1. Setiap komputer melakukan login dengan dua akun praktikan berbeda, satu akun pada peramban Mozilla dan akun lain pada Chrome. Sehingga jumlah akun pengguna yang digunakan sebanyak 20 akun.
2. Masing – masing akun membuat sebanyak 25 virtualisasi, namun dilakukan secara bertahap yaitu, lima virtualisasi setiap tahapan.
3. Masing – masing tahap dicatat berapa persentase memori RAM yang digunakan dan persentase penggunaan prosesor dengan melihat hasil monitoring dari aplikasi htop serta waktu respon aplikasi ketika menghidupkan virtualisasi terakhir dari masing-masing tahapan. Pencatatan memori RAM dan prosesor dilakukan untuk mengetahui apakah virtualisasi dapat berjalan semestinya dan dapat diakses ketika melakukan simulasi.

Sedangkan uji coba pada komputer pembanding berikut langkah – langkah dalam proses pengujian:

1. Skrip terdiri dari beberapa fungsi, yaitu membuat salinan dari virtualisasi yang sudah ada dan menghidupkan virtualisasi.
2. Fungsi yang dijalankan pertama kali adalah membuat salinan virtualisasi secara bertahap yaitu 10, 50, 100, 150, 200, dan 250 virtualisasi.
3. Kemudian setelah tahapan penyalinan selesai, fungsi kedua dijalankan dengan bertahap sama seperti langkah yang kedua.
4. Masing- masing tahap dicatat berapa memori RAM yang digunakan dan persentase dari prosesor yang digunakan dengan melihat hasil monitoring dari aplikasi htop.

Tabel 6 menampilkan hasil uji coba kapasitas dan performa pada komputer web server. Waktu respon yang ditampilkan merupakan hasil pencatatan waktu respon aplikasi ketika menghidupkan virtualisasi terakhir dari masing-masing tahapan.

Tabel 7 merupakan hasil uji kapasitas dan performa pada komputer pembeding. Virtualisasi maksimal yang berhasil dibuat adalah 150 dan 100 dalam kondisi hidup.

Tabel 6. Hasil Hasil Uji Kapasitas dan Performa pada Komputer *Web Server*

No	Jumlah Virtualisasi	Memori yang terpakai (8 GB)	Prosesor yang terpakai(4 core)	Waktu respon menghidupkan virtualisasi (ms)
1	100	22,48%	Core 1 : 8,2% Core 2 : 13,4% Core 3 : 12,2% Core 4 : 11,5%	826
2	200	36,83%	Core 1 : 22,8% Core 2 : 21,2% Core 3 : 23,2% Core 4 : 23,4%	1020
3	300	45,80%	Core 1 : 46,2% Core 2 : 44,0% Core 3 : 44,6% Core 4 : 50,8%	7510
4	400	55,13%	Core 1 : 51,7% Core 2 : 50,8% Core 3 : 50,5% Core 4 : 50,2%	12060
5	485	84,50%	Core 1 : 12,8% Core 2 : 11,9% Core 3 : 9,5% Core 4 : 13,2%	-

Tabel 7. Hasil Hasil Uji Kapasitas dan Performa pada Komputer Pembeding

No	Jumlah Virtualisasi	Memori yang terpakai(2 GB)	Prosesor yang terpakai(2 core)
1	10	9,6%	Core 1 : 0.0% Core 2 : 0.7%
2	50	37,4%	Core 1 : 3.2% Core 2 : 0.0%
3	100	47,7%	Core 1 : 0.0% Core 2 : 5.1%
4	150	61,6%	Core 1 : 12.3% Core 2 : 3.7%

D. Evaluasi

Berdasarkan hasil uji fungsionalitas, sistem dapat berjalan. Rata-rata waktu yang dibutuhkan untuk mengakses sekitar lima detik. Sedangkan berdasarkan hasil uji kapasitas dan performa, komputer *web server* dapat menangani virtualisasi sebanyak 400 ketika *web server* sedang digunakan oleh 20 pengguna dengan persentase memori RAM sebesar 55, 13%, prosesor rata-rata 50% dan waktu respon ketika menghidupkan virtualisasi mencapai sekitar 12 detik.

Modul praktikum jaringan komputer berupa *web server* sudah berhasil dilakukan oleh pengguna bernama Daniel Fablius dan Kharisma Nur Annisa menggunakan aplikasi simulasi jaringan ini.

Sedangkan minimal spesifikasi komputer yang dapat digunakan untuk menjalankan sistem adalah komputer dengan prosesor dual *core* dan memori RAM sebesar 2 GB. Dengan jumlah maksimal virtualisasi yang dapat dibuat sejumlah 100 virtualisasi, karena pada pengujian sebanyak 150 virtualisasi, komputer tidak dapat diakses lagi.

V. KESIMPULAN

Dalam proses pengerjaan Tugas Akhir yang melalui tahap perancangan, implementasi, serta uji coba, didapatkan kesimpulan sebagai berikut :

1. Aplikasi simulasi jaringan menggunakan Linux Container dan manipulasi paket menggunakan Scapy untuk praktikum jaringan komputer dapat diimplementasikan dengan waktu respon kepada pengguna rata-rata sekitar lima detik.
2. Komputer *web server* dapat menangani virtualisasi sebanyak 400 ketika *web server* sedang digunakan oleh 20 pengguna dengan persentase memori RAM sebesar 55, 13%, prosesor rata-rata 50% dan waktu respon ketika menghidupkan virtualisasi mencapai sekitar 12 detik.
3. Spesifikasi komputer yang dapat digunakan untuk menjalankan sistem adalah komputer dengan prosesor dual *core* dan memori RAM sebesar 2 GB. Dengan maksimal virtualisasi sebanyak 100 dengan persentase persentase yang digunakan RAM 47,7% dan prosesor 5,1%.

DAFTAR PUSTAKA

- [1] M. J. Scheepers, "Virtualization and Containerization of Application Infrastructure: A Comparison," University of Twente, Enschede, 2014.
- [2] Scapy, "secdev," secdev.org, [Online]. Available: <http://www.secdev.org/projects/scapy/>. [Accessed 6 April 2015].
- [3] J. Pan, "A Survey of Network Simulation Tools: Current Status and Future Developments," <http://www.cse.wustl.edu/>, Saint Louis, 2008.
- [4] linuxcontainers.org, "linuxcontainers.org," linuxcontainers.org, [Online]. Available: <https://linuxcontainers.org/lxc/introduction/>. [Accessed 6 April 2015].
- [5] sourceforge.net, "sourceforge.net," sourceforge.net, [Online]. Available: <http://lxc.sourceforge.net/man/lxc.html>. [Accessed 6 April 2015].
- [6] pocoo.org, "flask.pocoo.org," pocoo.org, [Online]. Available: <http://flask.pocoo.org/>. [Accessed 15 Oktober 2015].
- [7] Northwoods Software, "gojs.ne," Northwoods Software, [Online]. Available: <http://gojs.net/latest/index.html>. [Accessed 20 Oktober 2015].
- [8] linuxfoundation.org, "linuxfoundation.org," linuxfoundation.org, November 2009. [Online]. Available: <http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge>. [Accessed 6 April 2015].